Mid-level Features Improve Recognition of Interactive Activities

Kate Saenko
Ben Packer
C.-Y. Chen
S. Bandla
Y. Lee
Yangqing Jia
J.-C. Niebles
D. Koller
L. Fei-Fei
K. Grauman
Trevor Darrell

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-209
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-209.html

November 14, 2012

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
14 NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Mid-level Features Improve Recognition of Interactive Activities	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Electrical Engineering and Computer Sciences,Berkeley,CA,94720**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We argue that mid-level representations can bridge the gap between existing low-level models, which are incapable of capturing the structure of interactive verbs, and contemporary high-level schemes, which rely on the output of potentially brittle intermediate detectors and trackers. We develop a novel descriptor based on generic object foreground segments our representation forms a histogram-of-gradient representation that is grounded to the frame of detected key-segments. Importantly, our method does not require objects to be identi ed reliably in order to compute a ro- bust representation. We evaluate an integrated system including novel key-segment activity descriptors on a large-scale video dataset containing 48 common verbs, for which we present a comprehensive evaluation proto- col. Our results con rm that a descriptor de ned on mid-level primitives operating at a higher-level than local spatio-temporal features, but at a lower-level than trajectories of detected objects, can provide a substantial improvement relative to either alone or to their combination.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**17**	

Acknowledgement

This research was supported in part by DARPA's Mind's Eye Program.

Mid-level Features Improve Recognition of Interactive Activities

K. Saenko[1], B. Packer[2], C.-Y. Chen[3], S. Bandla[3], Y. Lee[3], Y. Jia[1], J. Niebles[2], D. Koller[2], L. Fei-Fei[2], K. Grauman[3], and T. Darrell[1]
[1]UC Berkeley, [2]Stanford, [3]UT Austin

Abstract

We argue that mid-level representations can bridge the gap between existing low-level models, which are incapable of capturing the structure of interactive verbs, and contemporary high-level schemes, which rely on the output of potentially brittle intermediate detectors and trackers. We develop a novel descriptor based on generic object foreground segments; our representation forms a histogram-of-gradient representation that is grounded to the frame of detected key-segments. Importantly, our method does not require objects to be identified reliably in order to compute a robust representation. We evaluate an integrated system including novel key-segment activity descriptors on a large-scale video dataset containing 48 common verbs, for which we present a comprehensive evaluation protocol. Our results confirm that a descriptor defined on mid-level primitives, operating at a higher-level than local spatio-temporal features, but at a lower-level than trajectories of detected objects, can provide a substantial improvement relative to either alone or to their combination.

1 Introduction

Broadly speaking, competing lines of research on activity recognition have focused on either "AI" based approaches, which exploit high-level models that involve explicit detection of objects, people and pose as an intermediate representation, or "learning" based models, which exploit low-level methods including point trajectories, local bag of feature models, etc. At the same time, empirical challenge problems that define the field have been progressing from relatively simple activities (e.g., *run, jump, walk*) to those that involve, complex, structured events and the interaction of multiple people and/or multiple objects (e.g., *exchange, hand-over, lead*). These latter "interactive" activities are most valuable for many real world applications, but have previously been the subject of relatively limited evaluation efforts.

Performance using low-level features and learning-based methods has been outstanding in many cases in earlier evaluations, but these new datasets provide

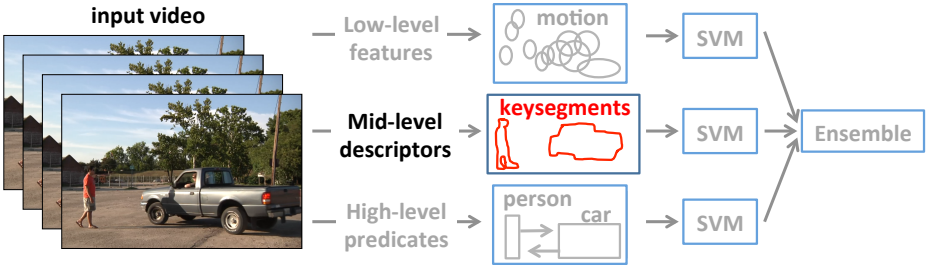

Figure 1: Mid-level descriptors based on generic object key-segment regions offer a trade-off between the robustness of low-level models and the structure captured by high-level models. We develop a new system which combines low-level motion features, mid-level key-segment descriptors, and high-level predicates on objects/people and their relationships, on a large-scale dataset containing 48 categories of interactive activities such as, in this example, *approach*.

a challenge: can low-level models really suffice when considering structured interactive activities? It would seem that rich intermediate representations must be essential for recognizing interactive verbs such as *open*, for example, given the broad range of instantiations of a semantic category (e.g. compare opening a door to opening a bottle).

Several authors have recently attempted to address this question in the case of static action/activity recognition [43, 25], motivated by the PASCAL activity recognition challenge [9]; however, our focus on the case of dynamic activities as revealed in video sequences, as in the classic activity recognition benchmarks of KTH, UCF Sports, Olympics, etc [18, 36, 28]. Dynamic interactive activity datasets–video corpora that include multiple people interacting in various roles–are more rare, and we focus our development on the recently introduced publicly downloadable "Visual Intelligence" (VISINT) activity dataset [1].

Recent results suggest that high-level models—those that operate on representations formed over tracked objects, object attributes, and/or interaction predicates between objects—are quite powerful at recognition of dynamic interactive activities [38, 5, 41]. A recent model for recognition based on interaction primitives [30] was shown to offer strong performance on the VISINT corpus; this model exploited appearance and STIP primitives defined on person and object trajectories obtained using a deformable part model detector [11]. Such models are powerful, but still fail to track all types of objects, or be robust to many types of observation conditions.

To improve robustness on complex activities that are difficult to track comprehensively using existing detectors, we introduce a novel mid-level activity descriptor based on generic object segments. We leverage the key-segments method of [22], and compute a descriptor based on static and dynamic properties of a detected key segment.

We combine this novel representation with two existing methods: a low-level latent state temporal sequence model [28], and a high-level model based on

a sequence of structured features including primitive representations of object state and person-object interaction [30]. We provide a comprehensive evaluation of these representations, separately and in combination, on the VISINT corpus. We show that the mid-level representation offers a clear improvement when combined with the previous techniques, and the best performance is obtained with the fusion of all three methods. Intuitively, we believe that the key-segments method improves in the cases where higher-level models are required (activities with structured interaction) but where the primitive trackers have failed to do object occlusion or appearance variation.

The dataset we use for evaluation is large and complex, and a further contribution of this paper is the protocols we designed for evaluating action recognition. In particular, humans do not always agree on verb presence when the verbs are defined most broadly; we propose a new metric for evaluating system agreement with noisy human labels. We hope this dataset, along with the protocol we developed, will be used by others to advance the state of the art in interactive activity recognition.

2 Background

Many researchers have adopted activity representations using low-level tracked points in a video as features [27, 29, 40, 39, 7, 20, 42, 24, 12]. Such a representation is prone to errors in tracking, which is especially true in the presence of background clutter, but it avoids the difficult task of object and person detection. A number of current approaches entail the use of local space-time interest points [39, 7, 29, 20, 4, 24, 42, 6, 26, 15]; several build representations using visual vocabularies computed with gradient-based descriptors extracted at detected interest points [7, 20, 6, 39, 42], while others build descriptors from the point positions themselves [4, 12]. The advantages of combining both static and dynamic descriptors have also been demonstrated [29, 24, 26, 15]. The strategy of generating compound neighborhood-based features—explored initially for static images and object recognition [44, 33, 21, 23, 32]—has since been extended to video [12, 6, 42, 20]. Various approaches either subdivide the space-time volume globally using a coarse grid of histogram bins [20, 6, 42, 15], or place grids around the raw interest points, and compute a new representation using the positions of the interest points that fall within the grid cells surrounding that central point [12].

Another line of work attempts to describe activities using intermediate predicates. At the mid-level, several approaches represent activities in terms of spatio-temporal shapes or segments [17, 3]. At the higher level, methods represent actions by the positions and velocities of an entire object using either a bounding-box detector [14] or a parts-based model [34, 35, 31, 13, 37]. Although a more intuitive framework, these representations suffer from the inherent inaccuracy of bounding box detection and from the fact that they do not model the entire scene and hence, cannot exploit contextual and geometric cues. On the other hand, the high-level approach adopted in our paper [30] can model activity

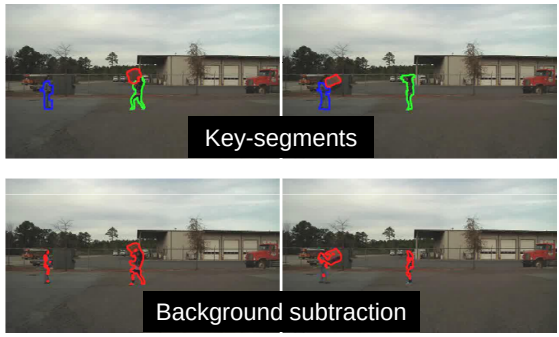

Figure 2: The key-segments output (top row) automatically generates space-time object segments that appear most central to the activity in the entire video clip. Unlike simple background subtraction (bottom row), they can distinguish the shapes of adjacent foreground objects, and extract object-like regions that do not move.

using both semantic and relational modules.

3 A Mid-level Activity Representation based on Key Segment Descriptors

A key challenge in realistic scenarios is that the system cannot know entirely in advance what objects may appear during an activity. While it is natural to search for people and an *a priori* bank of other very common objects, the system must be flexible enough to discover novel objects that appear and include them in the activity description. Thus, our approach to obtain a mid-level representation is to exploit multiple foreground segmentations, each corresponding to a unique human or object, but without having to detect or identify that object. To this end, we adapt a recent approach for *key-segment discovery* [22]. The method takes an unannotated video as input, and returns a ranked list of hypothesized space-time segmentations of the salient "object-like" regions as output. We use a *key-segments* decomposition of a video clip, which provides a space-time segmentation of the salient object-like regions that appear central to the activity [22].

Briefly, the key-segment extraction method we use works as follows: Given an unannotated video, we compute an initial pool of bottom-up regions. Then, we rank that pool according to how "human-like" and "object-like" each region appears. The former is based on a region's overlap with high-scoring person detections (we use [11]). The latter is based on the extent to which the region exhibits (1) appearance cues typical to objects in general (e.g., boundary strengths, probability of belonging to a vertical surface [8]), and (2) differences in motion patterns relative to its surroundings [22]. We cluster the top-ranked regions across all frames to form multiple key-segment hypotheses. Each hy-

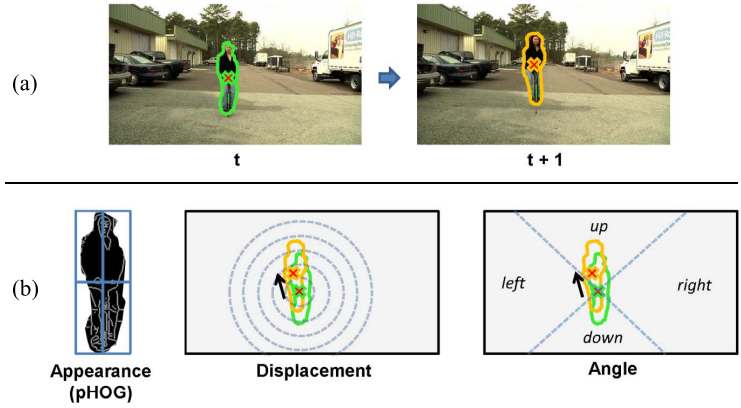

Figure 3: Overview of the mid-level segmentation descriptor. (a) Key-segments at frame t and $t+1$, where x's denote centroids. (b) Our descriptor encodes the segment's appearance (using quantized pHOG), its displacement, and its displacement angle in the next frame. We compute descriptors for each segment in all frames in the video; each increments a single bin in our final 3D appearance-motion histogram of the video. Best viewed in color.

pothesis defines a foreground color likelihood model within a space-time Markov Random Field (MRF), where each node is a pixel, and each edge connects adjacent pixels in space and time. We partition this graph with graph-cuts to obtain a pixel-wise segmentation for the discovered object/human as it moves over time. We compute the segmentations in order of hypothesis rank, and enforce non-overlap between the selected key-segments, such that each hypothesis corresponds to a unique human/object in the video. See Figure 2, and [22] for complete details.

With the key-segmentations in hand, now we want to describe each discovered object. This descriptor should capture the appearance and motion patterns, and ideally exploit the shape-based nature of the extracted segments (which contrasts with the cues a bounding box detector would provide). To this end, we design a novel mid-level descriptor that summarizes the shape of the object-like regions as well as their frame-to-frame motion trajectories, over the entire video clip. We process each space-time segmentation hypothesis separately, and then combine their features to create a single representation for the video.

To capture appearance, we compute a series of 2D pyramid of HOG (pHOG) descriptors on a window tightly fit to the segment, one for each frame where the segment appears. We compute the descriptor on a window that tightly fits the foreground segment in the frame, where the background pixels are zeroed out before the descriptor computation in order to capture the outer shape in addition to the internal contours. We then quantize the descriptors into 50 pHOG-words using k-means. To capture motion, we compute the difference in position and angle between the foreground segments in adjacent frames. We

quantize the positions into 10 bins, in 2 pixel increments from 0 pixels (i.e., $[0, 2), [2, 4), \ldots, [18, \infty)$); the bins range from small displacements to very large displacements. We quantize the angles into 4 bins, in $\pi/4$ increments from $\pi/4$; the bins correspond to up, down, left, or right (see Figure 3).

Using these measurements, we create a 3D histogram whose dimensions correspond to the appearance, distance, and angle, respectively. The size of the histogram is $50 \times 10 \times 4$. Each segment increments a single bin in the histogram. We aggregate the contributions of all segments in all space-time segmentation hypotheses to create a single histogram representation for the video. Finally, since some videos may generate no key-segment hypotheses due to missed person detections or high overlap in color distribution between the foreground and background models (which can lead to the foreground being "smoothed out"), we augment the mid-level descriptor with a histogram on the clip's space-time interest points and HoG/HoF features [20]. We train binary SVM classifiers using the resulting histograms to distinguish each verb against the rest.

4 An Activity Recognition System using Key-Segment Descriptors

We propose a multi-tier system design incorporating several levels of representation of increasing semantic richness. Our architecture is comprised of the novel mid-level description scheme defined in the previous section, as well as low- and high-level models based on previously reported methods. Our low-level representation employs a discriminative statistical sequence model built on top of sets of low-level spatio-temporal interest point (STIP) descriptors. Our high-level tier is a generative probabilistic sequence model incorporating high-level structural representations including person and object relationships. We combine the outputs of these components using a max-margin fusion scheme.

4.1 Low-level model

The algorithm implemented by our low-level tier is based on the method for activity classification described in [28]. This model is based on a framework for modeling motion by exploiting the temporal structure of human activities. The model represents activities as temporal compositions of motion segments, and a discriminative model is trained that encodes a temporal decomposition of video sequences, with STIP-based appearance models [18] for each motion segment. In recognition, a query video is matched to the model according to the learned appearances and motion segment decomposition. Classification is performed using a latent template max-margin model, based on the quality of matching between the motion segment classifiers and the temporal segments in the query sequence. The model is comprised of a set of motion segment classifiers each operating over a histogram of quantized interest points extracted from a temporal segment whose length is defined by the classifier's temporal scale. In addition to the temporal scale, each motion segment classifier also specifies a

temporal location centered at its preferred anchor point. Lastly, the motion segment classifier is enriched with a flexible displacement model that captures the variability in the exact placement of the motion segment within the sequence. [28] describes associated learning and inference procedures for this model.

4.2 High-level model

The high-level model evaluated here is based on the joint activity recognition and object tracking method of [30], which presents a model for understanding the interactions between humans and objects while performing an action over time. The model uses an "in the hand" interaction primitive and represents a variety of actions in which an object is manipulated. This representation not only allows for recognition of actions in sequences, but is also able to provide improved localization of the object of interest. The outputs of monocular object and person detectors are used as input to the Pose-Transition-Feature model of [30]. This model captures the spatial trajectory of the person and surrounding visual features, and is run on the person trajectory of a sequence to produce a score for each verb.

We then use the person and object trajectories and define three additional interaction primitives with respect to the object: "moving toward," "moving away from," and "touching". The primitives are observed in each frame, and we collect the counts of the primitives and transitions between the primitives in adjacent frames as additional features. For a given sequence, we concatenate the Pose-Transition-Feature scores with the primitive counts in addition to the following object-person interaction features: the percentage of frames each object-person primitive was active, the spatial variance of the object's trajectory, the spatial variance of the offset between the person and object, and an indictor variable for the identity of the object; the average object detection score in the sequence (to help the classifier ignore the object if it was not strongly detected). Using this feature vector, an SVM with a quadratic kernel is trained for each action separately.

4.3 Max-margin fusion

While the three individual models above have very different internal representations, they share a common max-margin learning framework, and thus are suitable to be fed into an ensemble pipeline. Specifically, we adopt a late fusion scheme by training a one-vs-all SVM with each of the low, mid, and high level features separately, and then combining their outputs via a soft voting scheme. To combine the outputs, we first convert the prediction of each SVM to a pseudo likelihood with a softmax function [2]:

$$p_{ci}(x) = \frac{\exp(m_{ci}(x))}{\sum_j \exp(m_{cj}(x))},$$

where m_{ci} is the output for class i from classifier c ($c \in \{L, M, H\}$). Our fused prediction is then based on a mixture of experts model [16]:

$$p_i^*(x) = \sum\nolimits_c w_c p_{ci}(x),$$

where the mixing coefficients $w_c (\sum_c w_c = 1)$ are found via cross-validation on the training data.

5 Data and Evaluation Procedure

5.1 Interaction Dataset

We based our evaluation on the publicly available VISINT dataset [1]; this dataset was recently collected to aid the development of action recognition methods, and represents a variety of commonplace interactions between humans by far exceeding that in previous datasets. Hired actors performed 10 exemplars of each action in outdoor scenes such as parks and streets, each 14 sec. long on average. Each exemplar action was shot 16 times, varying the dimensions: urban/park, daylight/evening, close/far field, center/side location within the frame. The full dataset was split into training and testing portions; a smaller subset containing fewer verbs was also used in some of our experiments. For the full set, we split the data to 3480 training and 1294 testing videos, and for the subset, we used 314 training and 86 testing videos. The dataset is provided with ground truth labels indicating per-video present/absent labels for each of the verbs in the corpus, which represent non-expert human responses from Amazon Mechanical Turk (AMT) to questions of the form *"Did you see X?"*, where X is one of the verbs, separately for each verb.

5.2 Computing Annotator Agreement

While care was taken to control present/absent verb label quality, this was mostly aimed at removing malicious workers and not at enforcing agreement. In fact, verb annotation in videos without specific annotator training besides providing the broad verb definition is a highly subjective task. Some annotators have an over-detection bias, answering "yes" to the question "Did you see X?" even if the action appeared briefly and was accompanied by many other actions. Others answer "yes" only if the action was central in the video sequence. Therefore, the resulting binary labels are noisy and not reliable sources of training data for traditional binary discriminative classifiers.

Fortunately, the dataset contains 16 variations per *exemplar*, or the same action (see Section5.1). This allows us to combine the human responses for the 16 unique videos that represent the variants of an exemplar action, effectively resulting in a score from 0 to 16 for that action for each of the 48 verbs[1].

[1] A few videos that did not have all 16 variants were regarded as not having a label and removed from evaluation.

Agreement	*Positive*	*Negative*	*Use in Dataset*
50%/75%	≥8/16 said yes	≤4/16 said yes	subset, full
62%/87%	≥10/16 said yes	≤2/16 said yes	subset, full
93%/93%	≥15/16 said yes	≤1/16 said yes	full

Table 1: Levels of agreement used in our experiments to map Turker votes into binary labels.

verb	BOUNCE	HAVE	HIT	HOLD	KICK	LEAVE	TOUCH	WALK	Total
yes(≥10)	12	16	12	16	16	16	33	13	134
no(≤2)	37	12	29	28	41	20	16	36	219

Table 2: The number of labels at 62%/87% level of agreement in the training portion of *subset*.

To obtain binary present/absent labels from the tallied votes, we tried several agreement thresholds, in order of increasing conservatism: (1) treat as positive videos for which 8 or more of the 16 annotators said the verb was present, and as negative those where 4 or fewer said the verb was present; (2) treat as positive those with 10 or more "yes" votes, and treat as negative those with 2 or fewer; and (3) treat as positive those with 15 or 16 votes, and as negative those with at most 1 vote. These are summarized in Table 1. The most stringent level did not produce enough labels in the *subset* datasets (10 or more positive and negative per verb). A list of verbs in the subset and their number of *yes* and *no* labels at the second level of agreement is shown in Table 2.

5.3 Evaluation Metrics

We use both a traditional detection metric (mean average precision) to evaluate w.r.t. binary labels, and propose a divergence-based metric to capture how well the predicted likelihood of an action agrees with the distribution of human judgments.

Mean Average Precision (mAP): mAP is traditionally used to evaluate detections of binary labels, and is better-suited to unbalanced data than accuracy. The AP for a binary detection problem is defined as the average precision obtained by varying a threshold (sensitivity) of the classifier, where precision is the number of true positives divided by the total number of assigned positive labels at a particular threshold. The mAP is then defined as the mean AP across all binary problems (across all verbs in our case).

Mean JS Divergence (mJSD): mAP is limited to hard binary labels and cannot measure the distance between soft labels (probabilities) of each verb being present, as given by the human votes. It is important to note that unlike most object recognition tasks, the verb labels for action recognition are not mutually exclusive, e.g., a set of responses for a video may have $p(\text{“}go\text{”}) = 0.9$ and $p(\text{“}walk\text{”}) = 0.9$, thus computing a distance between the overall distribution of responses for all verbs is incorrect, as it does not constitute a probability distribution of a single random variable, but rather a set of distribu-

tions of several random variables corresponding to the presence or absence of each verb. Thus, we propose to compute the distance separately for each label, using the Jensen-Shannon divergence (JS-divergence). Let Q denote the Bernoulli distribution of a verb being present given the human response data, and P denote the system response distribution, both of which have been normalized. Then the KL-divergence is given by $D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$. JS-divergence compares the two distributions (P and Q) to their mean as follows: $D_{JS}(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M)$. JS-divergence has a number of advantages over the KL-divergence: (a) it can handle zero probabilities, and is less sensitive to small numerical values; (b) it is symmetric and thus a true distance; (c) it is bounded in $[0, 1]$, whereas KL-divergence is unbounded. Finally, the mean JS divergence (mJSD) metric for a test video is calculated by averaging over verbs.

5.4 Procedure

The following sections describe the details of the experimental setup specific to each system component, such as the processing applied to the videos, which inputs are given to the system at each level, and the model parameters used in the experiments.

Low-level model settings. We subsample all videos to a size of 640×360. We use spatio-temporal interest points detected with the 3D Harris corner method [18] and described with HOG/HOF descriptors, using the binaries available at [19]. We run the detector on each video with the parameters: `-res 2`. If no detections are found, (e.g. resolution is too low to capture the motion in the video), we run the detector again with the parameters: `-res 4`. We then uniformly sample 200 videos from the training set and use all their local feature descriptors to form a codebook. The codebook is computed using k-means with $k = 500$. Finally, we train binary classifiers for each verb separately using agreement labels, using a fixed number of $K = 3$ motion segments per model.

Mid-level model settings: For efficiency, we generate the initial region pool on 40 frames uniformly sampled from the video. We compute up to four segmentations per video (two human, two other generic objects). For the color-likelihood models we use 5 fg and 10 bg GMMs and the RGB color space. We normalize the histograms to sum to 1, and use χ^2 kernels for the SVMs, with $C = 100$.

High-level model settings: Object and person detections for the model were computed using the DPM model [10]; a Viterbi tracker was run on top of the person detections to provide a trajectory for the primary person of interest in the sequence. We selected which single object was present in the sequence according to which object had the highest maximum detection score in each frame, averaged across frames. We included a second person as a possible object, only using detections that were outside of the trajectory found for the primary person. A Viterbi tracker was similarly run on the chosen object to produce an object trajectory. Codewords of STIP features from the low-level model were

used as local appearance features near the person track.

5.5 Results

In addition to results obtained by different levels of representation, we also show several baselines: assigning random probabilities to the verbs (*Random*), assigning the prior probability as measured on the training set (*Prior*), and a baseline k-nearest neighbor classifier with $k = 1$ (*Baseline*). The latter retrieves the training examples with the nearest STIP feature vectors to the input video and returns the averaged human verb distributions.

Effect of Agreement: First, we investigate the effect of annotator agreement on classifier performance. Figure 4 shows the mAP score obtained by *Baseline* and *LowLevel*, as well as the random baselines, using labels obtained by requiring increasing levels of agreement (Table 1). The trend is that the stricter agreement produces more accurate results, however, the strictest level (15/16 or 93% for *yes* and *no*, i.e. all but one must agree) results in a smaller set of available training labels, producing lower accuracy. The second level (62% for *yes* and 87% for *no*) is therefore optimum, and we only report results with these labels for the rest of the paper. Note that, because each agreement level produces different sets of binary labels, the number of verbs that have sufficient labeled positive and negative examples changes: e.g., at the 50%/75% level, 47 verbs had sufficient (10 or more) positive and negative examples in the training set.

Activity Recognition: We first compare the performance of each level of representation on *subset*, which at the 62%/87% agreement contains sufficient labels for: *bounce, have, hit, hold, kick, leave, touch,* and *walk*. These verbs were chosen as a representative set, and the high-level relational primitives were designed with these verbs in mind. Table 3 (columns labeled "subset") shows the mAP and mJSD obtained by the models. Overall, results are encouraging: all models achieved scores well above chance performance and significantly higher than those of *Baseline*. The highest single-component mAP is obtained by the high-level model (0.75). We believe its good performance is explained by its use of interaction features. Figure 5 compares the performance by verb in terms of the mAP score. Here we see the complementariness of the "pixel" vs. "predicate" approaches: the high-level model does significantly better on verbs *bounce, have, hit* and worse on *touch,walk*. In particular, its poor performance on *walk* can be explained by the fact that *walk* does not involve interaction between humans and objects.

Finally, we evaluate performance on the full dataset. With the 62%/87% agreement labels, this amounts to 46 verbs (all but *move* and *bury*). The results are shown in Table 3 (columns labeled "full"). Here the low-level model outperforms the others; the lower performance of the high-level model may be because the primitives used are not enough to capture the other actions beyond the eight verbs they were designed for (extending the high-level primitives to more verbs is part of future work). Finally, we combine all three representation levels, obtaining the best overall results: mAP=.81, mJSD=.0397 on the

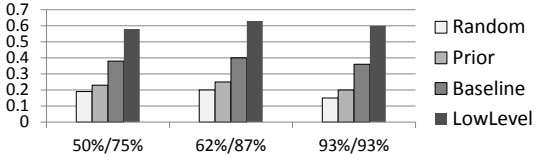

Figure 4: Comparison of verb recognition mAP at increasing levels of annotator agreement.

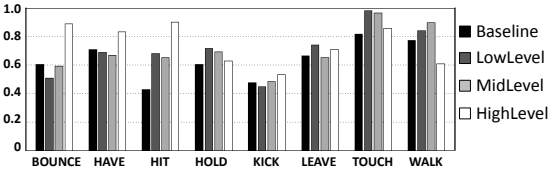

Figure 5: Breakdown by verb of mAP obtained on the subset, using annotator agreement 62%/87%.

subset and mAP=.71, mJSD=.0198 on the full data dataset. The fact that the combined model performs significantly better than either level alone indicates that there is additional information in the low-level features that is not being exploited by the intermediate mid- and high-level models.

Our evaluation in terms of the divergence between the predicted verb probability and the human judgements reveals that the lowest divergence is also obtained by the combined three-level scheme. However, the behaviour of this metric is different from mAP. Notice that using the verb prior to predict labels (*Prior*) only improves mAP marginally, but cuts the mJSD to a third on the full dataset. This suggests that the prior distribution of certain verbs (e.g. *go* is likely to be present in most videos) may play a larger role in cases where humans do not agree.

6 Conclusion

Based on our results, we argue that intermediate representations should be used in addition to low-level features to get best performance; one reason high-level models fail is they throw away useful information available in the sequence by committing to the (possibly erroneous) object track. We presented a novel mid-level representation based on generic object key-segments found in video sequences; our approach combined elements of low and high-level representations. While the key-segments approach alone was not the strongest model, in concert with the other paths it significantly improved performance.

	mAP		mJSD	
model	subset	full	subset	full
Random	.50	.20	.1099	.1086
Prior	.59	.25	.0639	.0301
Baseline	.63	.40	.0623	.0270
Low	.70	.63	.0489	.0286
Mid	.70	.59	.0600	.0259
High	.75	.49	.0594	.0361
Low+High	.75	.69	.0544	.02541
Low+Mid+High	**.81**	**.71**	**.0397**	**.0198**

Table 3: Results obtained by the different representation levels on the interaction dataset, using annotator agreement for binary labels of 62%/87%. The table shows mAP (higher is better) and mJSD (lower is better) scores.

References

[1] `http://www.visint.org/`.

[2] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[4] M. Bregonzio, S. Gong, and T. Xiang. Recognizing action as clouds of space-time interest points. In *CVPR*, 2009.

[5] W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. In *CVPR*, 2011.

[6] J. Choi, W. Jeon, and S.-C. Lee. Spatio-temporal pyramid matching for sports videos. In *ACM Multimedia*, 2008.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[8] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[10] P. Felzenszwalb, D. Girshick, and D. R. McAllester. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[11] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[12] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009.

[13] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29(2):2247–2253, 2007.

[14] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots - learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[15] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[16] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[17] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *CVPR*, 2007.

[18] I. Laptev. On space-time interest points. *IJCV*, 64(2):107–123, 2005.

[19] I. Laptev. *Space-Time Interest Points*, 2010. Software available at `http://www.irisa.fr/vista/Equipe/People/Laptev/download.html`.

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[21] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning from Partially Matching Images. *International Journal of Computer Vision (IJCV)*, 85(2), May 2009.

[22] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[23] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. In *ICCV*, 2007.

[24] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[25] S. Maji, L. Bourdev, and J. Malik. Action recognition using a distributed representation of pose and appearance. In *CVPR*, 2011.

[26] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[27] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009.

[28] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[29] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[30] B. Packer, K. Saenko, and D. Koller. A combined pose, object, and feature model for action understanding. In *CVPR*, 2012.

[31] V. Parameswara and R. Chellappa. Human action-recognition using mutual invariants. *CVIU*, 1998.

[32] D. Parikh, L. Zitnick, and T. Chen. Unsupervised learning of hierarchical spatial structures in images. In *CVPR*, 2009.

[33] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.

[34] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[35] C. Rao and M. Shah. View-invariance in action recognition. In *CVPR*, 2001.

[36] M. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[37] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[38] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *CVPR*, 2009.

[39] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[40] E. Shechtman and M. Irani. Space-time behavior based correlation or how to tell if two underlying motion fields are similar without computing them? *PAMI*, 29(11):2045–2056, 2007.

[41] M. Sridhar, A. Cohn, and D. Hogg. Unsupervised learning of event classes from video. In *AAAI*, 2010.

[42] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[43] B. Yao, X. Jiang, A. Khosla, A. Lin, L. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

[44] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *CVPR*, 2007.